Санкт-Петербургский политехнический
университет Петра Великого
Институт компьютерных наук и технологий
Кафедра компьютерных систем и программных технологий

Отчёт по лабораторной работе
Дисциплина: Телекоммуникационные технологии
Тема: Сигналы телекоммуникационных технологий

Выполнил студент группы 33501/3 _______________ П.М.Шувалов
(подпись)

Преподаватель _______________ Н.В.Богач
(подпись)

Санкт-Петербург
2017

Сигналы телекоммуникационных систем

1 Цель

Познакомиться со средствами генерации и визуализации сигналов.

2 Постановка задачи

В командном окне MATLAB и в среде Simulink промоделировать сигналы из Главы 3, сс. 150–170 (см. Справочные материалы).

3 Теоретический раздел

А.Б. Сергиенко Цифровая обработка сигналов: Глава 1,сс.18–25, Глава 3, сс. 150–170.

В работе используются простейшие навыки работы с MATLAB. Данный отчет по сути является теоретическим пособием, вся необходимая информация приведена в каждом конкретном примере ввиду их простоты.

4 Ход работы

Первой исследуемой функцией будет затухающая синусоида. Ниже представлен код для ее генерации.

Код MATLAB:

```
Fs = 8e3;     % частота дискретизации 8 кГц
t = 0:1/Fs:1; % одна секунда дискретных значений времени
t = t';       % преобразуем строку в столбец
A = 2;                          % амплитуда - два вольта
f0 = 1e3;                       % частота - 1 кГц
phii = pi/4;                    % начальная фаза 45
s1 = A * cos(2*pi*f0*t + pi); % гармонический сигнал
alpha = 1e3;                    % скорость затуханий
s2 = exp(-alpha*t).*s1;         % затухающая синусоида
```

Другие функции, которыми удобно использовать для представления дискретных сигналов:

- Рис. 1, a – график, строимый plot по умолчанию;
- Рис. 1, b – отображаются отсчеты;
- Рис. 1, c – сигнал в виде «стебельков»;
- Рис. 1, d – график в ступенчатом виде.

Команды для построения графиков:

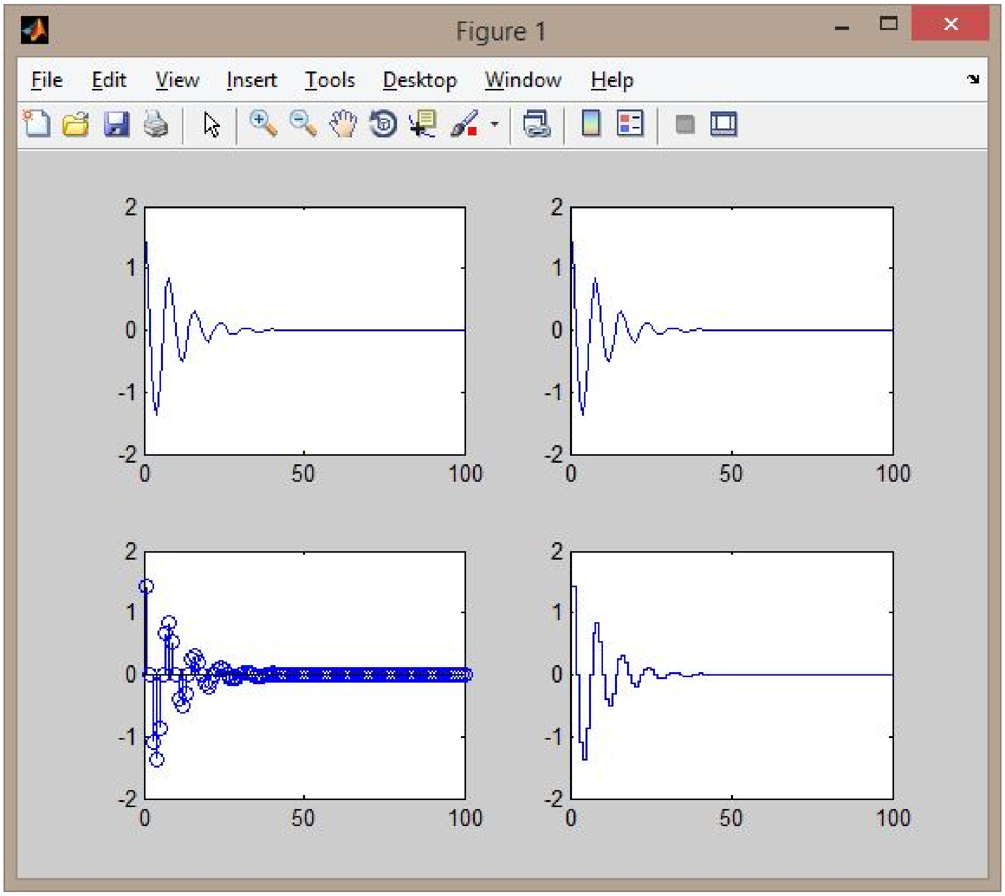

Рис. 1: Дискретный сигнал на графиках.

Код MATLAB:

```
subplot(2, 2, 1); plot(s2(1:100));      % привычный график
subplot(2, 2, 2); plot(s2(1:100)); % отсчеты
subplot(2, 2, 3); stem(s2(1:100));      % стебельки
subplot(2, 2, 4); stairs(s2(1:100));    % ступеньки
```

Горизонтальная ось на графиках Рис. 1 пронумеромана номерами отсчетов. Нужно передать и временной вектор и второй, тогда пронумернованы будут обе оси: `plot(t(1:100), s2(1:100))` (Рис. 2).

Генерация многоканального сигнала на Рис. 3:

Код MATLAB:

```
f = [600 800 1000 1200 1400]; % вектор частот (строка!)
s3 = cos(2*pi*t*f);           % пятиканальный сигнал
figure(3)
plot(t(1:100), s3(1:100, :));
```

В результате `s3` – матрица, содержащая значения произведения времени и частоты. Итог вычислений показан на Рис. 3:

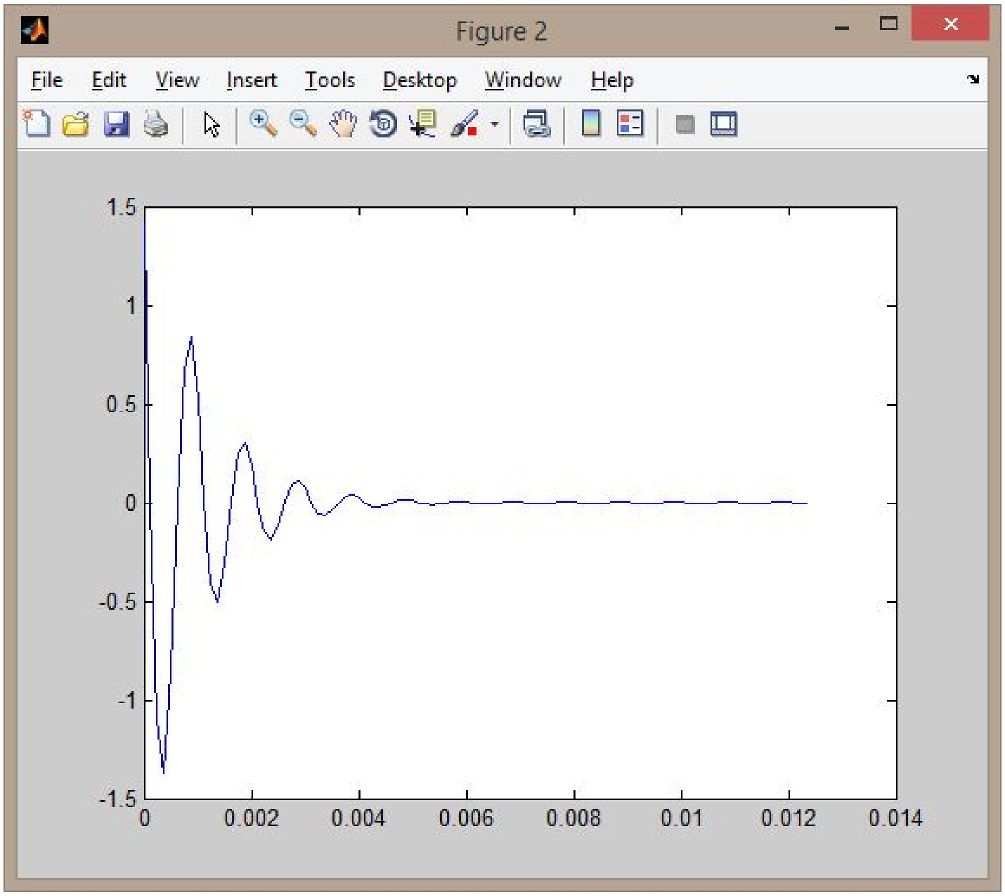

Рис. 2: Градуированная временной ось.

4.1 Кусочные зависимости

Создадим кусочные функции, которые на разных участках задаются разными формулами.

Используем операцию сравнения, она зануляет выражение, когда FALSE. на Рис. 4.

Разберем построение одностороннего экспоненциального импульса:

Код MATLAB:

```
s4 = A*exp(-alpha*t).*(t >= 0);
```

Прямоугольный импульс, центрированный относительно начала отсчета времени:

Код MATLAB:

```
Fs = 100;
t = -1:1/Fs:1; % 2 секунды дискретных значений времени
T = 0.5;
s5 = A*(abs(t) <= T/2);
```

Несимметричный треугольный импульс:

Код MATLAB:

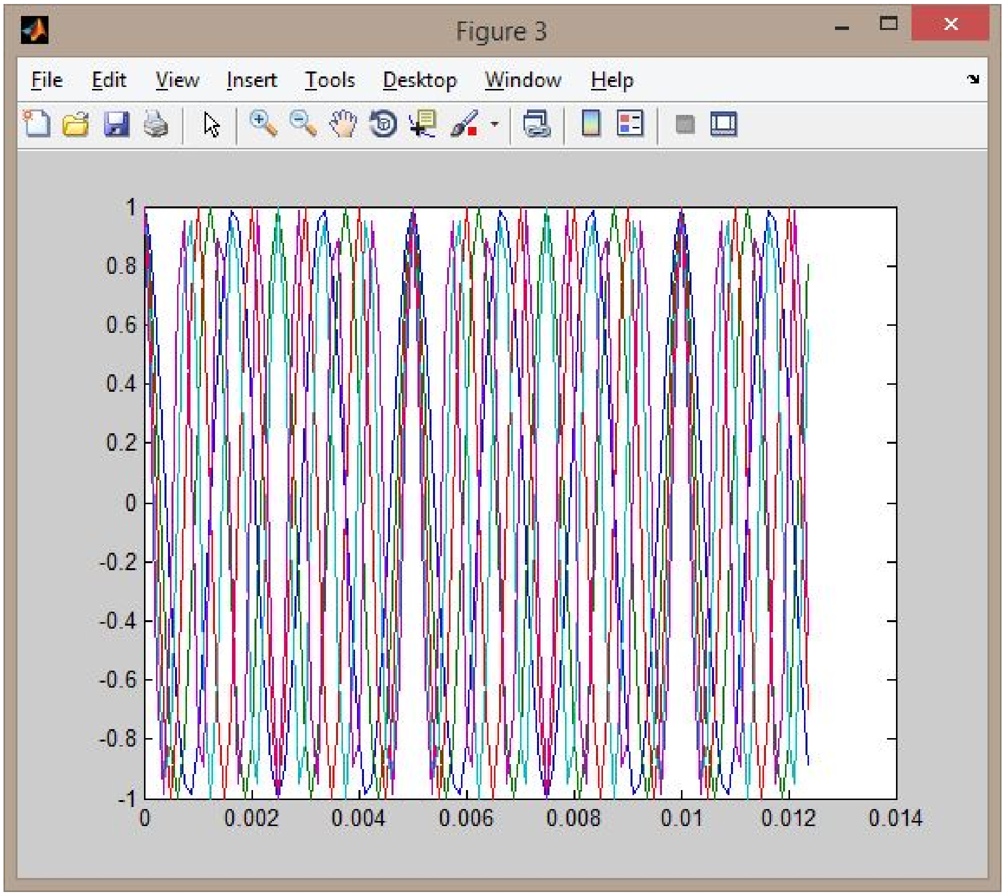

Рис. 3: Многоканальный сигнал.

```
s6 = A*t/T.*(t >= 0).*(t <= T);
```

Также можно воспользоваться другим способом задания кусочных зависимостей: выполнить вычисления только для тех моментов времени, для которых действительно необходимо:
Односторонний экспоненциальный импульс:

Код MATLAB:

```
% заполняем вектор сигнала нулями
s4 = zeros(size(t));
% находим номера неотрицательных элементов вектора t*f
inds = find(t >= 0);
% рассчитываем сигнал только в нужных точках
s4(inds) = A * exp(-alpha * t(inds));
```

Прямоугольный импульс, центрированный относительно начала отсчета времени:

Код MATLAB:

```
s5 = zeros(size(t));
s5(find(abs(t) <= T/2)) = A;
```

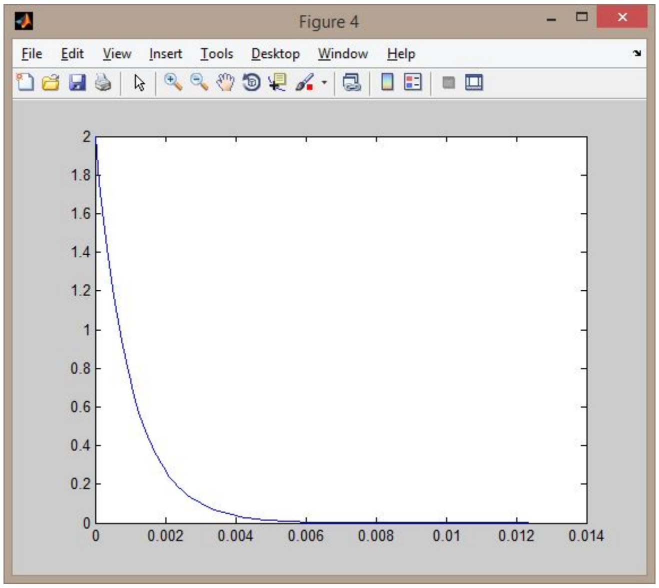

Рис. 4: Экспоненциальный сигнал

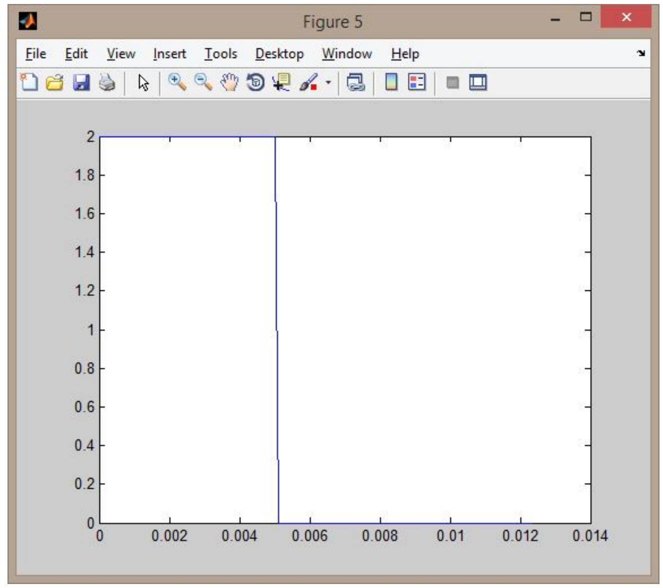

Рис. 5: Прямоугольный сигнал

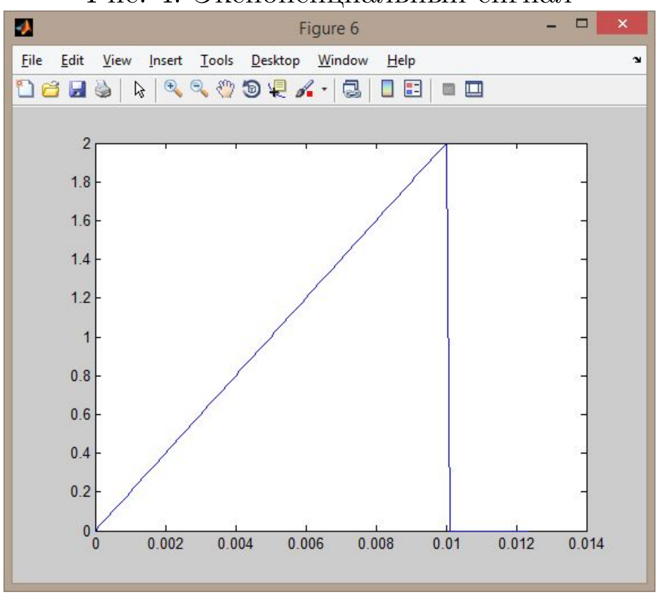

Рис. 6: Несимметричный треугольный импульс

Несимметричный треугольный импульс:

Код MATLAB:

```
s6 = zeros(size(t));
inds = find((t>=0) & (t <= T));
s6(inds) = A*t(inds)/T;
```

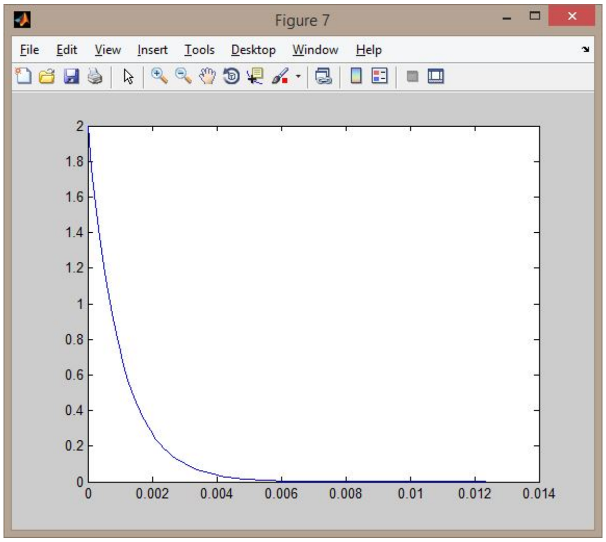

Рис. 7: Экспоненциальный сигнал 2

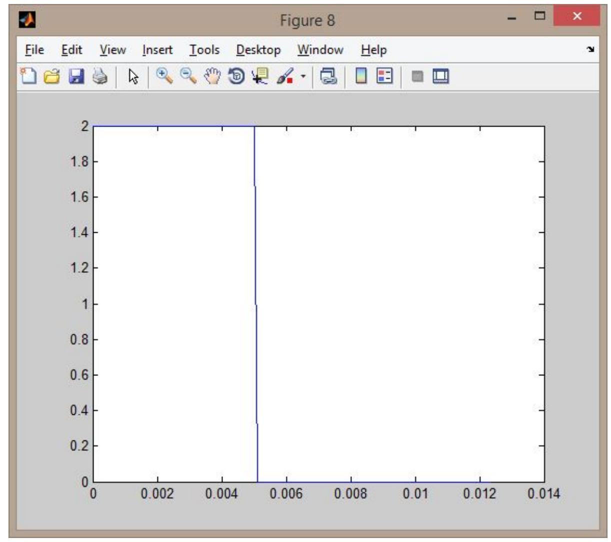

Рис. 8: Прямоугольный сигнал 2

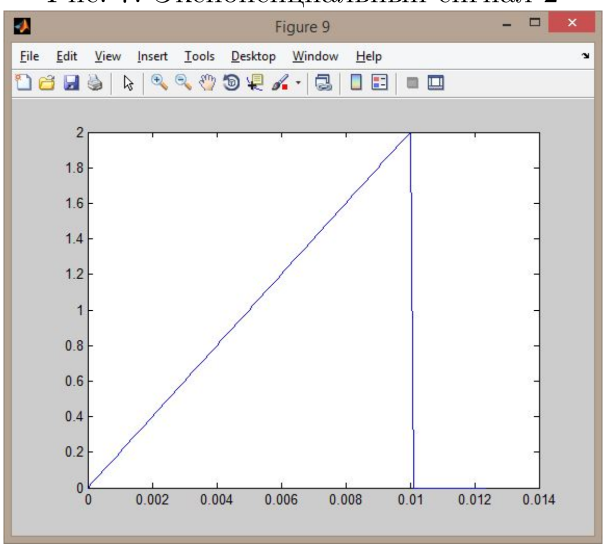

Рис. 9: Несимметричный треугольный импульс 2

4.2 Прямоугольный импульс

Одиночный прямоугольный импульс: `y = rectpuls(t, width)`. `t` – вектор значений времени, `width` – ширина импульса. Возвращаемый результат `y` – вектор рассчитанных значений сигнала, определяемый формулой:

$$y = \begin{cases} 1, -\frac{width}{2} \leq t \leq \frac{width}{2}, \\ 0, t < -\frac{width}{2}, t \geq \frac{width}{2}. \end{cases} \tag{1}$$

Создание прямоугольных импульсов с амплитудой 5 В и длительностью 20 мс каждый, расположенных справа и слева от начала отсчета времени. Результат показан на Рис. 10:

Код MATLAB:

```
Fs = 1e3;                % частота дискретизации
```

```
t = -40e-3:1/Fs:40e-3; % дискретное время
T = 20e-3;             % длительность импульсов
A = 5                  % амплитуда
s = -A*rectpuls(t+T/2, T)+A*rectpuls(t-T/2,T);
plot(t,s);
ylim([-6 6]);
```

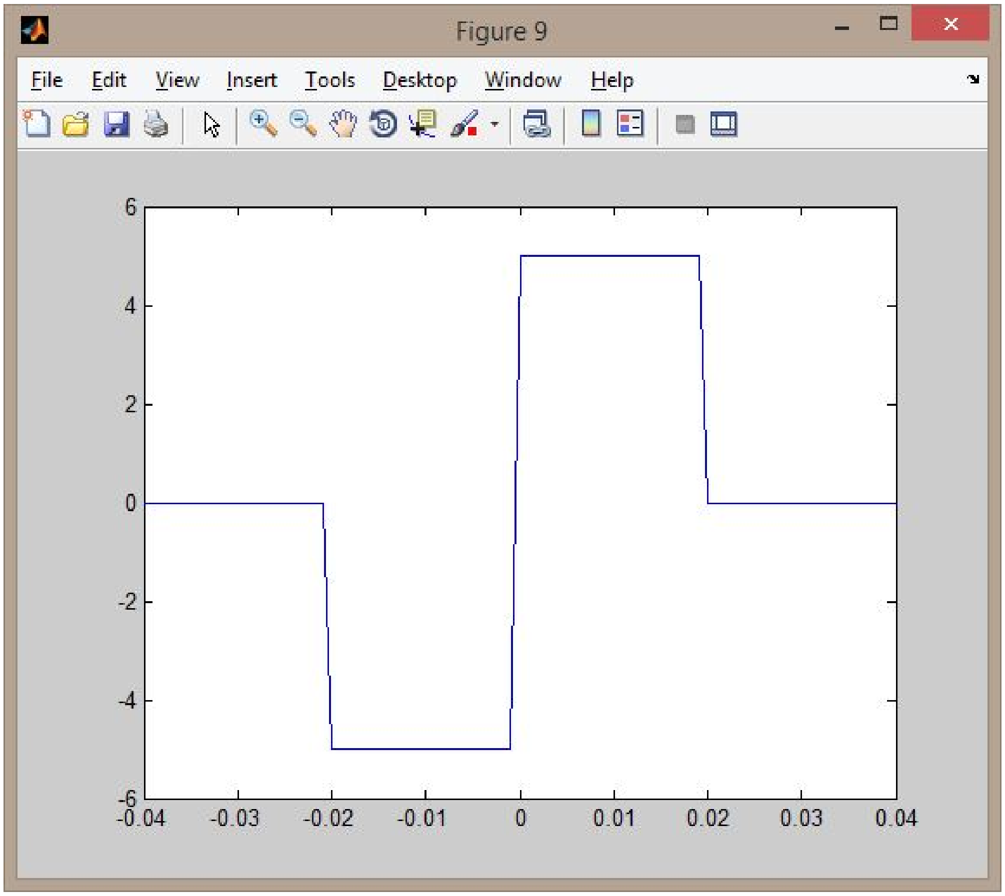

Рис. 10: Прямоугольный сигнал rectpuls.

4.3 Треугольный импульс

Одиночный треугольный импульс с единичной амплитудой: `y = tripuls(t, width, skew)`. `t` – вектор значений времени, `width` – ширина импульса, `skew` – коэффициент асимметрии импульса, определяющий положение его вершины. Пик импульса расположен при `t=width*skew/2`. Параметр `skew` должен лежать в диапазоне от -1 до 1. Формула:

$$y = \begin{cases} \frac{2t+width}{width(skew+1)}, -\frac{width}{2} \leq t < \frac{width \cdot skew}{2}, \\ \frac{2t-width}{width(skew-1)}, \frac{width \cdot skew}{2} \leq t < \frac{width}{2}, \\ 0, |t| > \frac{width}{2}. \end{cases} \tag{2}$$

Приведу пример формирования симметричного трапециевидного импульса с амплитудой 10 В и размерами верхнего и нижнего оснований 0 и 50 мс соответственно. Результат показан на Рис. 6:

Код MATLAB:

```
Fs = 1e3;
t = -50e-3:1/Fs:50e-3;
A = 10;
T1 = 20e-3;
T2 = 60e-3;
s = A*(T2*tripuls(t,T2)-T1*tripuls(t,T1))/(T2-T1);
plot(t,s);
```

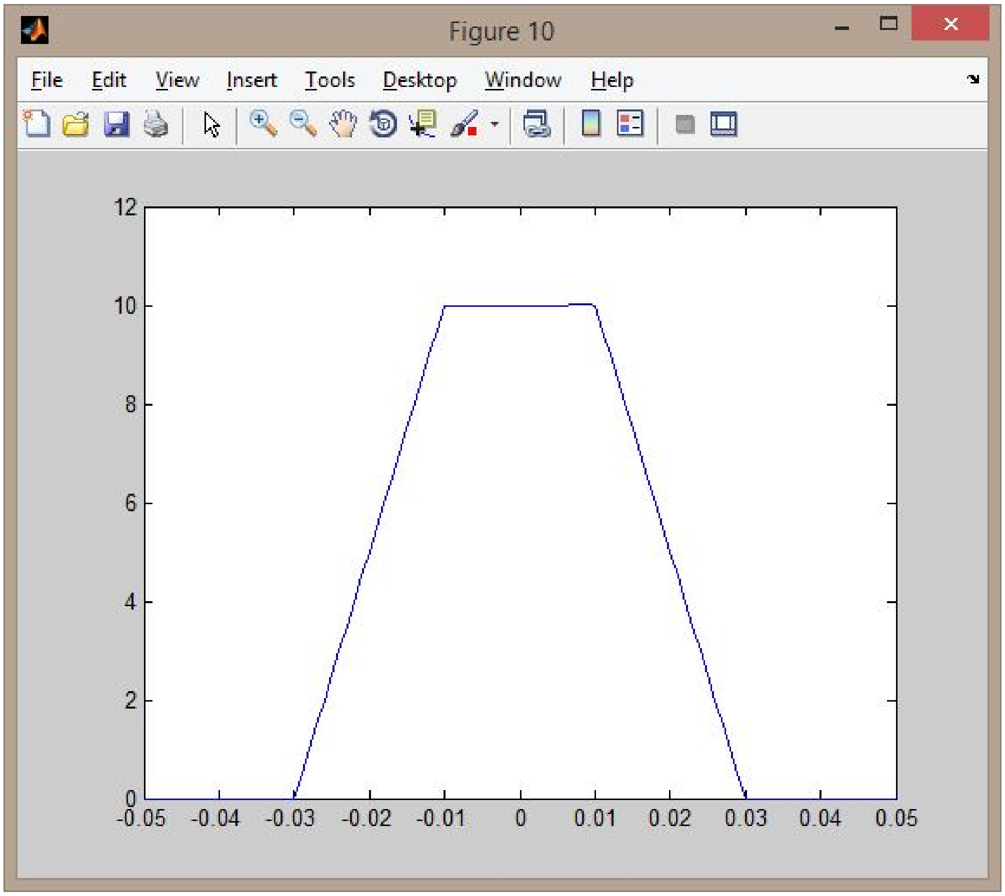

Рис. 11: Сигнал, сформированный с помощью tripuls.

4.4 Гауссов радиоимпульс

Одиночный радиоимпульс с гауссовой огибающей и единичной амплитудой: `y = gauspuls(t, fc, bw, b` `t` – вектор значений времени, `fc` – несущая частота в герцах, `bw` – относительная ширина спектра, `bwr` – уровень в децибелах, по которому производится измерение ширины спектра. Формула:

$$y = \exp(-at^2)\cos(2\pi f_c t). \tag{3}$$

Гауссов радиоимпульс с несущей частотой 4 кГц и относительной шириной спектра 10%, измеренной по уровню -20 дБ. Результат на Рис. 7:

Код MATLAB:

```
Fs = 16e3;              % частота дискретизации
t = -10e-3:1/Fs:10e-3; % дискретное время
Fc = 4e3;               % несущая частота
bw = 0.1;               % относительная ширина спектра
bwr = -20;              % уровень измерения ширины спектра
s = gauspuls(t, Fc, bw, bwr);
plot(t, s);             % график сигнала
ylim([-1.1 1.1]);
```

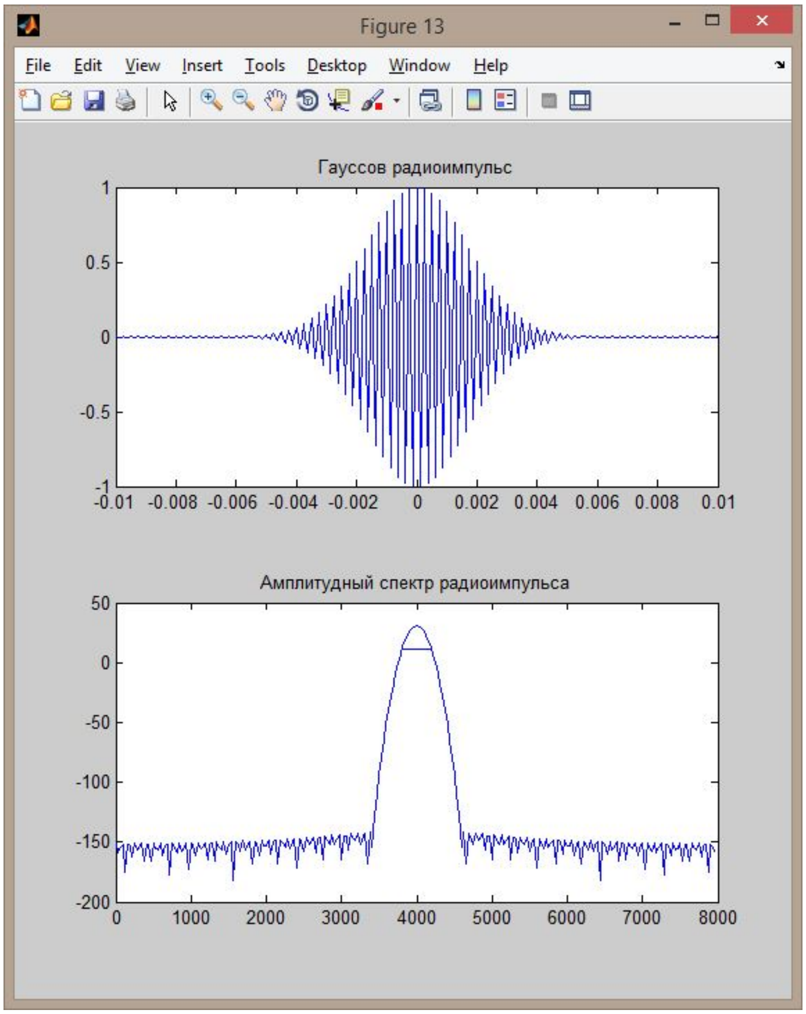

Рис. 12: Гауссов радиоимпульс и его спектр

4.5 Генерация последовательности импульсов

Последовательность из пяти симметричных треугольных импульсов, интервалы линейно увеличиваются, амплитуды экспоненциально уменьшаются. Результат на Рис. 4.4.

Код MATLAB:

```
Fs = 1e3;                        % частота дискретизации
t = 0:1/Fs:0.5;                  % дискретное время
tau = 20e-3;                     % длительность импульса
d = [20 80 160 260 380]'*1e-3;   % задержки импульсов
d(:,2) = 0.8.^(0:4)';            % амплитуды импульсов
y = pulstran(t,d,'tripuls',tau);
plot(t, y)
```

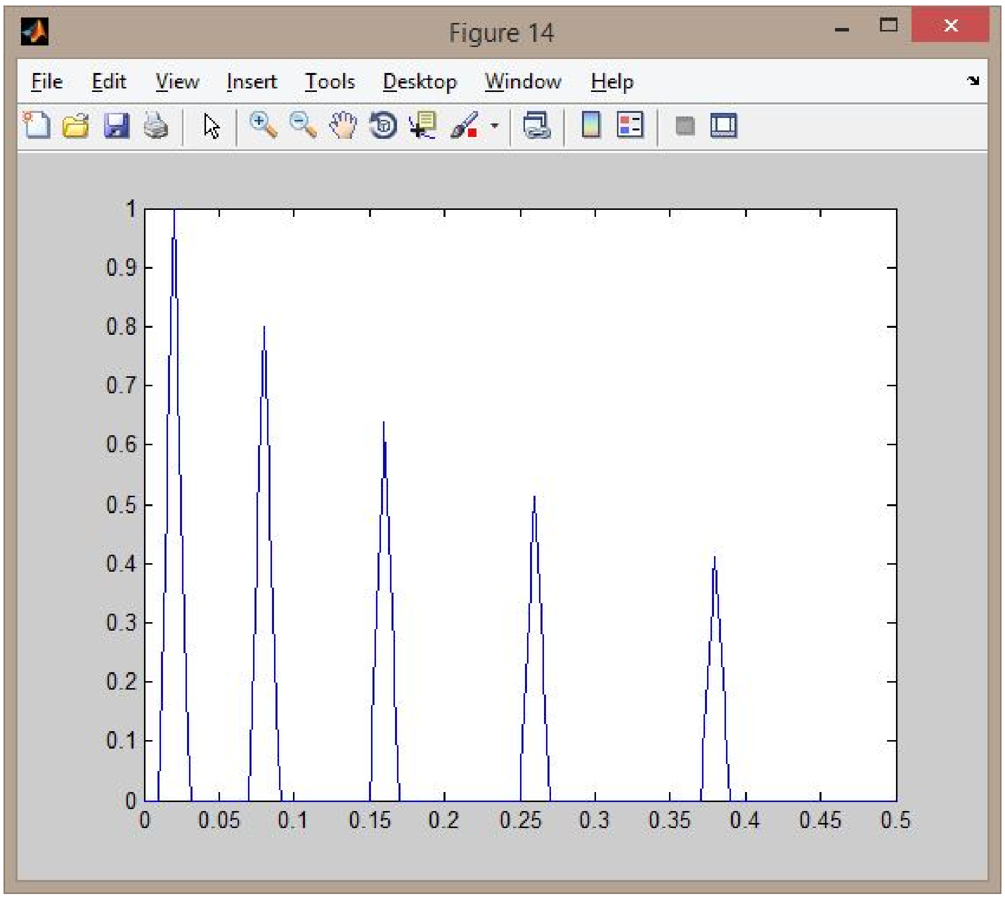

Рис. 13: Последовательность pulstran

4.6 Последовательность треугольных импульсов

Последовательность с периодом T формируется так: $y = sawtooth(2\pi * t/T)$.

Результат на Рис. 13.

Код MATLAB:

```
Fs = 1e3;                  % частота дискретизации
t = -25e-3:1/Fs:125e-3; % дискретное время
A = 5;                     % амплитуда
T = 50e-3;                 % период
t1 = 5e-3;                 % длительность падающего участка
s = (sawtooth(2*pi*t/T, 1-t1/T)-1)*A/2;
plot(t,s)
```

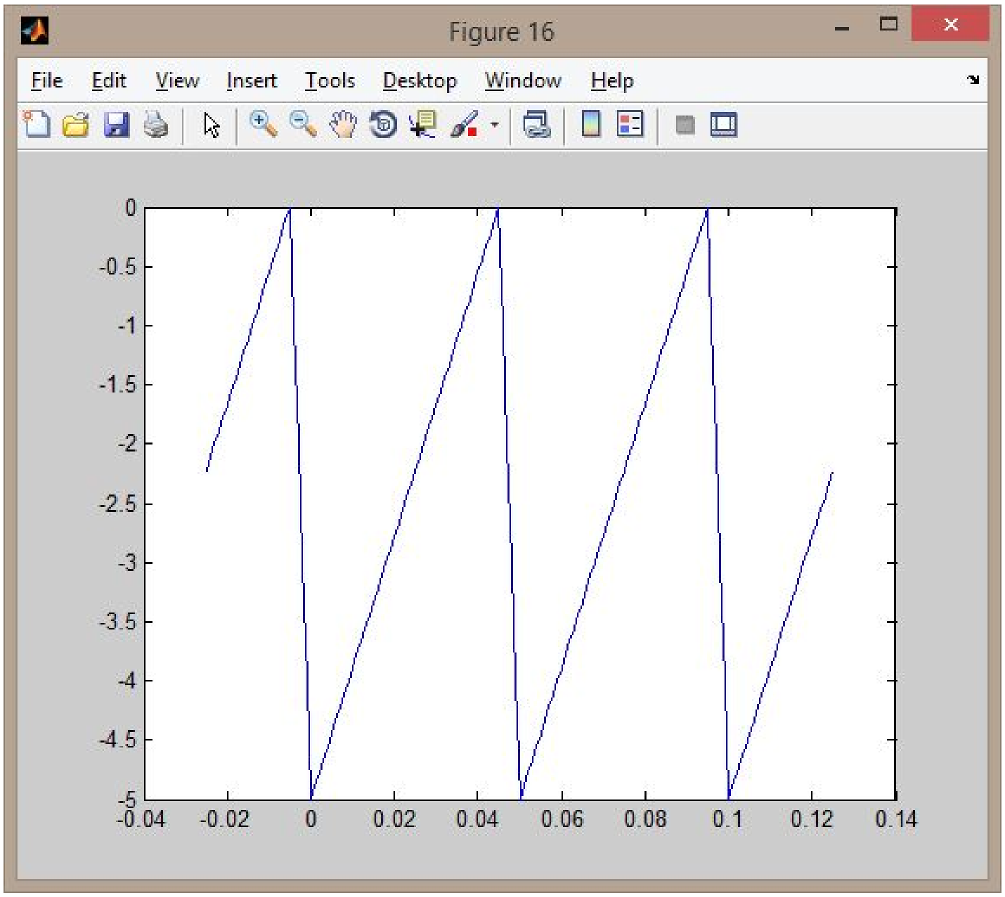

Рис. 14: Sawtooth.

4.7 Функция Дирихле

Функция Дирихле:

$$\mathrm{diric}_n(x) = \sum_{k=-\infty}^{\infty} \mathrm{sinc}\left(n\left(\frac{t}{2\pi} - k\right)\right), \tag{4}$$

Функция MATLAB: `y=diric(x,n)`, `x` и `n` .

Графики функции Дирихле при нечетном значении n - Рис. 15.

Код MATLAB:

```
x = 0:0.01:15;
```

```
subplot(2,1,1);
plot(x, diric(x, 7));
grid on;
title('n = 7');
subplot(2,1,2);
plot(x, diric(x, 8));
grid on;
title('n = 8');
```

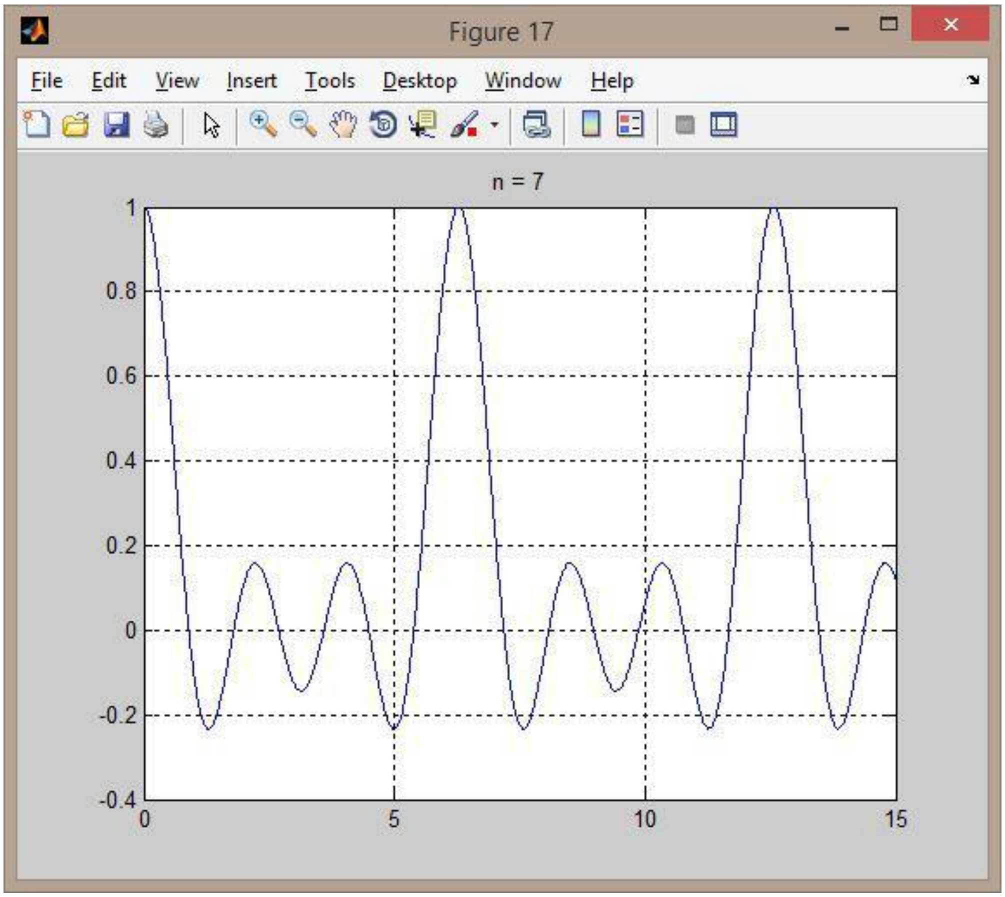

Рис. 15: Функция Дирихле

5 Выводы

В результате выполнения работы приобрели навыки генерации и визуализации сигналов в MatLab. Сигналы можно разделить на детерминированные и случайные. При выполнении работы промоделировали основные виды детерминированных сигналов. Случайные сигналы образуются под действием случайных физических процессов, поэтому предсказать их можно только с определенной вероятностью. Для анализа систем используются детерминированные сигналы. Сигналы можно разделить на непрерывные и дискретные. Детерминированные сигналы делятся на периодические(гармонические и полигармонические) и непериодические.